Requirements for an expert system explanation facility

Requirements for an expert system explanation facility

JOHANNA D. MOORE

Department of Computer Science and Learning Research and Development Center, University of Pittsburgh, Pittsburgh, PA 15260, U.S.A.

AND

CÉCILE L. PARIS

Information Sciences Institute, University of Southern California, 4676 Admiralty Way, Marina del Rey, CA 90292-6695, U.S.A.

Received August 9, 1991

Revision accepted September 24, 1991

Comput. Intell. 7, 367-370 (1991)

For the past several years, we have worked on building an explanation component for an expert system building framework (or "shell"), the Explainable Expert System (EES) Framework. In this short paper, we describe the characteristics that we believe to be essential for an explanation component of an expert system. We then identify important features of the EES architecture that support the desired capabilities. Finally, we discuss some areas where fruitful work remains to be done.

1. Characterisitcs of a good explanation facility

Every type of natural language application has its own requirements. Based on studies of human explanation giving and on our system-building experience, we outline here the requirements we have identified for an expert system explanation facility.

- *Fidelity.* In order to be trusted, an explanation must accurately reflect the system's knowledge and reasoning (Swartout 1983, 1990). To ensure fidelity, an explanation facility must synthesize text directly from the same knowledge sources used for problem solving. It cannot rely on templates or canned text written a priori by a programmer because, as the system evolves, it is impossible to ensure that the corresponding text is an accurate reflection of the program's behavior.
- *Knowledge from multiple sources.* To support the range of questions users wish to ask, an expert system must provide several different knowledge sources, including terminological knowledge, factual domain knowledge, problem-solving knowledge, and an execution history (see Swartout *et al.* 1991). An explanation facility must have strategies that enable it to *decide* on the type(s) of information needed and *extract* it from these knowledge sources.
- *Naturalness.* Expert systems are often required to produce multisentential explanations, for example, to provide justifications of the system's actions or recommendations, descriptions of its problem-solving strategies, or definitions of the terms it uses. The explainer must thus be able to *organize* information into a coherent presentation. To generate coherent natural language explanations, the system's explanations should follow standard patterns of discourse employed by humans. To do so, the explanation facility must have knowledge about discourse structure and strategies for employing that knowledge.

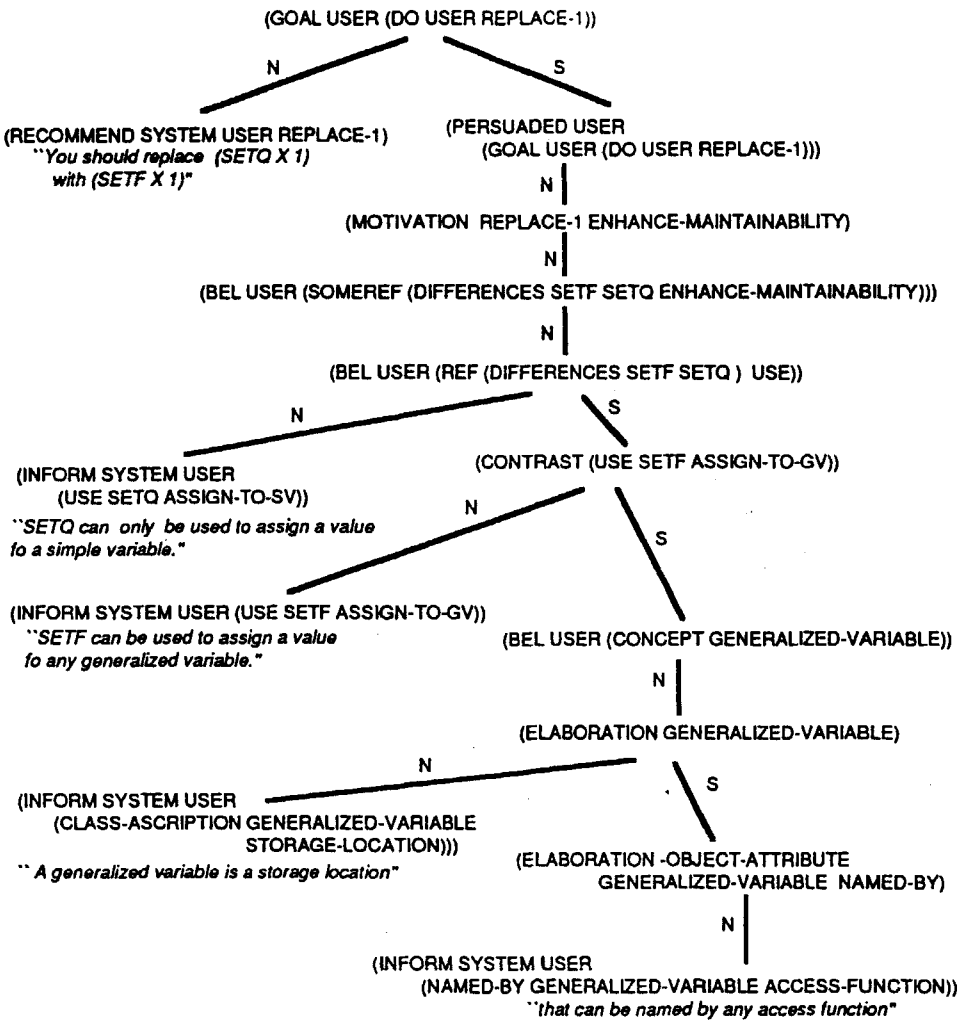
The naturalness criterion applies not only to individual explanations, but to entire explanation dialogues as well. Taken as a whole, the explanations produced by the system during a dialogue must form a coherent set.

- *Responsiveness.* Analyses of naturally occurring advisory dialogues show that advice-seekers often do not understand the advisor's explanations and frequently ask follow-up questions (Moore 1989*b*). The system must thus be able to answer follow-up questions or offer an alternative explanation if a user is not satisfied with a given explanation.
- *Flexibility.* An explanation facility must have a variety of strategies for answering a given question. This is important for at least two reasons. First, if a given explanation is not understood, the system must be able to offer an alternative explanation. Second, the system must be able to present the same information from different perspectives, depending on contextual factors such as the user's knowledge or goals.
- *Sensitivity.* Explanations should be influenced by information about the user's knowledge and goals, the problem-solving situation, and the previous dialogue. Question and answer pairs cannot be treated independently; later explanations must take prior explanations into account.
- *Extensibility.* An explanation facility must be able to handle many different types of questions, some of which may not be foreseen at design time. It must be possible to easily extend the explanation facility to respond to additional question types or to add new strategies for answering existing question types.
- *Portability.* Because our generation facility is a component of an expert system shell, it can include only domain-independent strategies. The architecture must thus aid users in customizing the explainer by adding domain-dependent strategies to suit their requirements.
- *Adaptive capabilities.* An explanation facility should have the ability to automatically modify its existing strategies and learn new strategies based on experience and interactions with users (Paris 1990).

2. Brief system overview

We have devised an explanation architecture that explicitly plans explanations to achieve discourse goals denoting what information should be communicated to the user (e.g., make the user know a certain concept, persuade the user to perform an action). When a goal is posted, the planner searches its library of explanation strategies looking for candidates that could achieve it. In general, there will be several candidate strategies for achieving a goal. The planner employs

Printed in Canada / Imprimé au Canada



Sample explanation generated by this text plan:

> You should replace (SETQ X 1) with (SETF X 1). SETQ can only be used to assign a value to a simple-variable. SETF can be used to assign a value to any generalized-variable. A generalized-variable is a storage location that can be named by any access function.

FIG. 1. A completed text plan and the explanation produced.

a set of *selection heuristics* to determine which strategy is most appropriate in the current situation. These selection heuristics take into account information about the user's knowledge and goals (as recorded in the *user model*), the conversation that has occurred so far (as recorded in the *dialogue history*), and information about whether or not a strategy requires assumptions to be made. Once a strategy is selected, it may in turn post subgoals for the planner to refine. Planning continues in a top-down fashion until all goals are refined into speech acts, such as INFORM and RECOMMEND.

As the system plans explanations, it records the goal structure of the response being produced. In addition, it keeps track of any assumptions it makes about what the user knows, as well as alternative strategies that could have been chosen at any point in the planning process. The result is a *text plan* for achieving the original discourse goal. Text plans are recorded in the dialogue history before being passed to the Penman text generation system (Mann and Matthiessen 1983), which performs the process of realization into English text.

In our system, a completed text plan is more than simply a specification for the realization process. It is an explicit representation of the planning or "design" process that produces an explanation. To briefly illustrate this point, we show a sample text plan in Fig. 1 accompanied by the text it produces. This text plan comes from an interaction with the program enhancement advisor (PEA) (Neches *et al.* 1985), an advice-giving system constructed using the EES framework that is intended to aid users in improving their Common Lisp programs.[1]

The system produces the text plan shown in Fig. 1 to satisfy the discourse goal of achieving the state in which the user has adopted the domain goal of replacing (SETQ X 1)

[1]PEA recommends transformations that improve the "style" of the user's code. It does not attempt to understand the content of the user's program.

with (SETF X 1), denoted by (GOAL USER (DO USER REPLACE-1)) in the figure. Basically this text plan does the following. It recommends that the user performs the replacement and then attempts to persuade the user to do this act. To persuade the user to replace SETQ with SETF, it motivates this act in terms of the shared domain goal of enhancing the maintainability of the program. To motivate this act, the system then chooses a strategy that contrasts the object being replaced (SETQ) with the object replacing it (SETF) in terms of the shared domain goal ENHANCE-MAINTAINABILITY. Although there are many differences between SETQ and SETF in the expert system's knowledge base, the difference relevant to the current domain goal of enhancing maintainability is in the generality of their usage. Thus, the system informs the user that SETQ can be used to assign a value to a simple variable and contrasts this with the use of SETF, namely to assign a value to any generalized variable. Finally, as the planner constructs the speech act to inform the user that SETF may be used to assign a value to any generalized variable, it determines from the user model that it must introduce a concept, GENERALIZED-VARIABLE, that is not known to the current user. This causes the planner to post the additional subgoal (BEL USER (CONCEPT GENERALIZED-VARIABLE)) to introduce this new term. This goal is achieved by providing the elaborating information defining the concept in terms of its class membership and defining attributes.

As shown in Fig. 1 and described more fully in Moore and Paris (1989), a text plan represents the roles individual clauses in the text play in achieving discourse goals, as well as how the clauses relate to one another rhetorically. For example, contrasting the usage of SETQ with the usage of SETF is intended to achieve the discourse goal of persuading the user to do the replacement act. Moreover, the portion of text that contains the contrast is rhetorically related to the recommendation by a MOTIVATION relation. Similarly, defining the term GENERALIZED-VARIABLE is intended to make the user know this concept, and this text serves as ELABORATION for the INFORM act which first introduces the term. In addition, information about what entities are salient at each point in the explanation (attentional information) can be derived from a text plan.

3. Important architectural features

In our system, knowledge about explanation is represented explicitly in a set of plan operators that were derived by studying naturally occurring explanations. These operators integrate multiple sources of knowledge. First, they encode standard ways that discourse (i.e., intentional) goals are achieved by rhetorical means, thus achieving our goal of *naturalness*. For example, as shown in the figure above, one strategy persuades the user to perform a replacement act using the rhetorical strategy of motivating the act by contrasting the replacee with the replacer in terms of achieving shared goal(s). Second, operators contain applicability constraints that specify the knowledge that must be available if the operator is to be used. These criteria can refer to the expert system's knowledge bases, the user model, or the dialogue history. For example, the operator for explaining a concept by analogy requires that there be an analogous concept that is familiar to the user or has been mentioned previously in the dialogue. Planning using these operators allows the system to produce coherent explanations directly from the expert system's domain knowledge (*fidelity*). Because operator constraints also reference the user model, the system can tailor the content and organization of its explanations to the individual user (*sensitivity*) (Paris 1990; Moore and Paris 1992).[2]

A second important feature of our architecture is that explanations themselves are structured objects that the system can reason about. We have demonstrated that the text plans recorded by our system provide the context necessary for handling a range of dialogue phenomena. More specifically, by reasoning about the prior explanations it has produced, our system is able to interpret users' follow-up questions and answer them in context of the ongoing dialogue (Moore and Swartout 1989; Moore 1989*b*), select a perspective when describing or comparing objects (Moore 1989*a*), and avoid repeating information that has already been communicated or that the user already knows (Moore and Paris 1989). Thus our system exhibits *responsiveness* and some types of *sensitivity* to dialogue context.

4. Future directions

While we have made considerable progress towards achieving our goals, there are several areas in which much work remains. First, there is considerable discussion about the type of planning architecture that is best suited for explanation generation. Some argue for an approach in which an explanation is completely planned and revised by critics before it is generated (Suthers 1991; Lester and Porter 1991). Others advocate a simple goal refinement strategy which facilitates the interleaving of planning with realization (Cawsey 1989; Moore 1989*a*). There is a clear tradeoff here. The former approach allows explanations to be optimized along various dimensions, but must expend considerable effort planning and revising without feedback from the user. The latter approach allows incremental explanation generation and thus the developing explanation can adapt to the changing context quickly as user feedback is received. This tradeoff should be further explored through empirical tests.

Yet others argue that planning text requires several inherently different types of reasoning, and therefore that a single top-down planning mechanism is insufficient (Hovy 1988; Mooney *et al.* 1991; Suthers 1991). We believe that it is crucial to clearly identify the range of explanation tasks and the control strategies best suited to achieving each of these tasks. Suthers (1991) has begun this process.

Another issue that concerns us is the feasibility of providing domain-independent explanation strategies, which we must do if we are to provide a shell for constructing explainable expert systems. EES has already been used to construct expert systems in several different domains. In addition to the program enhancement advisor, EES has been used to construct expert systems for diagnosing faults in a space station, diagnosing faults in local area networks, and aiding physicians in prescribing treatment for patients in the cardiac intensive care unit. We have been able to reuse a considerable number of explanation operators across these various domains. These include operators for justifying the system's actions, defining terms, describing objects, and describing the system's general problem-solving strategies. While it is

[2]In addition, Bateman and Paris (1989) have begun to examine how to tailor the phrasing of the generated texts to the users.

likely that additional domain-specific strategies will be needed in some domains (Kittredge *et al.* 1991), our initial experience is promising.

Finally, we believe that more work must be done in understanding the nature of explanation dialogues. Explanation dialogues are largely user-controlled, and thus the structure of the dialogue emerges only as the interaction proceeds. We have only begun to understand this structure and the processes that access it. Many interesting questions remain. How should the dialogue history be managed and used? Can things be forgotten (i.e., removed from the dialogue history)? Which things and when? How should later explanations be affected by previous utterances? What information must be recorded in order to guarantee that, taken as a whole, the explanations produced by the system form a coherent set? What is the relationship between the dialogue history and user model? Are these separate entities? If so, should information migrate from the dialogue history to the user model? What information? How can we handle other dialogue phenomena such as interruptions and subdialogues? Do we require meta-level strategies for managing the dialogue, such as the *discourse plans* of Litman and Allen (1987) or those of Cawsey (1989)? We hope to be able to study some of these questions in the future.

Acknowledgment

The research described in this paper was supported by the Defense Advanced Research Projects Agency (DARPA) under a NASA Ames cooperative agreement number NCC 2-520.

BATEMAN, J.A., and PARIS, C.L. 1989. Phrasing a text in terms the user can understand. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit, MI, pp. 1511–1517.

CAWSEY, A. 1989. Generating explanatory discourse: a plan-based interactive approach. Ph.D. thesis, University of Edinburgh, Edinburgh, United Kingdom.

HOVY, E.H. 1988. Two types of planning in language generation. Proceedings of the Twenty-Sixth Annual Meeting of the Association for Computational Linguistics, State University of New York, Buffalo, NY, pp. 179–186.

KITTREDGE, R., KORELSKY, T., and RAMBOW, O. 1991. On the need for domain communication knowledge. Computational Intelligence, 7(4): this issue.

LESTER, J., and PORTER, B. 1991. An architecture for planning multi-paragraph pedagogic explanations. Proceedings of the AAAI-91 Workshop on Comparative Analysis of Explanation Planning Architectures, Anaheim, CA, pp. 27–41.

LITMAN, D.J., and ALLEN, J.F. 1987. A plan recognition model for subdialogues in conversations. Cognitive Science, 11: 163–200.

MANN, W.C., and MATTHIESSEN, C. 1983. Nigel: a systemic grammar for text generation. Technical Report RR-83-105, Information Sciences Institute, University of Southern California, Marina del Rey, CA.

MOONEY, D.J., CARBERRY, M., and MCCOY, K.F. 1991. Capturing high-level structure of naturally occurring extended explanations using bottom-up strategies. Computational Intelligence, 7(4): this issue.

MOORE, J.D. 1989*a*. A reactive approach to explanation in expert and advice-giving systems. Ph.D. thesis, University of California, Los Angeles, CA.

_____ 1989*b*. Responding to "huh?": answering vaguely articulated follow-up questions. Proceedings of the Conference on Human Factors in Computing Systems, Austin, TX, pp. 91–96.

MOORE, J.D., and PARIS, C.L. 1989. Planning text for advisory dialogues. Proceedings of the Twenty-Seventh Annual Meeting of the Association for Computational Linguistics, Vancouver, B.C., pp. 203–211.

_____ 1992. User models and dialogue: an integrated approach to producing effective explanations. *In* User modeling and user-adapted interaction. Kluwer Academic Publishers, Dordrecht, The Netherlands. In press.

MOORE, J.D., and SWARTOUT, W.R. 1989. A reactive approach to explanation. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit, MI, pp. 1504–1510.

NECHES, R., SWARTOUT, W.R. and MOORE, J.D. 1985. Enhanced maintenance and explanation of expert systems through explicit models of their development. IEEE Transactions on Software Engineering, **SE-11**(11): 1337–1351.

PARIS, C.L. 1990. Generation and explanation: building an explanation facility for the explainable expert systems framework. *In* Natural language generation in artificial intelligence and computational linguistics. Kluwer Academic Publishers, Boston/Dordrecht/London. pp. 49–82.

SUTHERS, D.D. 1991. Task-appropriate hybrid architectures for explanation. Computational Intelligence, 7(4): this issue.

SWARTOUT, W.R. 1983. XPLAIN: a system for creating and explaining expert consulting systems. Artificial Intelligence, 21(3): 285–325. Also available as Report ISI/RS-83-4, Information Sciences Institute, University of Southern California, Marina del Rey, CA.

_____ 1990. Evaluation criteria for expert system explanation. Proceedings of the AAAI-90 Workshop on Evaluation of Natural Language Generation Systems, Boston, MA.

SWARTOUT, W.R. PARIS, C.L., and MOORE, J.D. 1991. Design for explainable expert systems. IEEE Expert, 6(3): 58–64.